Interdisciplinary Development Patterns from the use case: Support-U

Christian Voigtmann[1], Matthias Söllner[2],
Daniel Wilhelm[1], Klaus David[1] and Jan Marco Leimeister[2]

[1]Chair for Communication Technology, [2]Information Systems
University of Kassel
Pfannkuchstraße. 1, 34121 Kassel
{voigtmann, soellner, wilhelm, david, leimeister}@uni-kassel.de

Abstract: Software development has proven to be a challenge. To address this challenge, there are quite many interesting approaches how to develop software - starting from the waterfall approach, up to recently quite popular agile software development techniques. Another already some years old approach was described by the Gang of Four and proposes the usage of design patterns to provide a general reusable solution to commonly occurring problems in software development. Although design patterns have been around for a long time, their usability is still promising. To the best of our knowledge "interdisciplinary patterns" to address challenges in the development of context aware application in ubiquitous environments have not been described in literature so far. Hence, this paper proposes and also evaluates concrete interdisciplinary software development patterns. To provide an application example the proposed patterns are used to address two use cases that commonly occur in the development process of context aware applications: providing transparency to the user and ensuring a user's self-determination. For the demonstration of the patterns Support-U a context aware application that provides elderly people to live autonomously is used.

1 Introduction

In many of today's products, such as cars, and in many of today's procedures such as the optimization of business processes, E-Government or telecommunication and its services, software is becoming more and more important. The development of software has been - and still is - quite a challenge. To address, this challenge a few different structured approaches how to develop software have been introduced by the discipline of software engineering. One "classical" approach is the "waterfall" model [IR09]. Other approaches are the V-Model [SL08], the spiral model, or agile software programming methods such as scrum or extreme programming. Yet another approach is "software design patterns". One important reference for this approach is given by Gamma et al. [GHJV95]. The idea of patterns are to provide reusable "software entities", which are tested, solve important problems and help to reach the final goal of a running software system faster and more efficiently. Although this approach is almost two decades old, it is still an important approach today. Starting with the seminal visionary paper from Mark Weiser [Wei91],

Pervasive Computing or Ubiquitous Computing is becoming more and more reality toady and therefore become more important. This is mainly due to the fact that smartphones are omnipresent and becoming more capable to be used in ubiquitous computing systems with respect to their computing capability, their networking capability and especially with respect to their sensing capabilities [DJ12]. Moreover, in addition to touch screens the human computer interaction - already "foreseen" by Mark Weiser - "to become invisible to the user" are further enhanced and connected using speech interfaces which are getting more and more in the focus e.g. of Google Glasses [Man]. The design and development of ubiquitous systems is a research focus of the interdisciplinary R&D project Venus. In this project various disciplines of computer science together with economics, jurists, and usability experts are co-operating. Based on several years of work in this project, we present an approach to design and develop such ubiquitous systems to be more socially acceptable by using what we call "interdisciplinary patterns". To the best of our knowledge, there are no publications about interdisciplinary patterns for the development of context aware applications in ubiquitous systems so far. The need for these interdisciplinary patterns arises both from the use cases outlined in this paper as well as from the non technical challenges as mentioned above.
This paper is organized as follows: Section 2 presents the Support-U application whose development has been affected by the interdisciplinary patterns. Section 3 outlines the underlying architecture of the ubiquitous environment where Support-U is applied. Section 4 introduces the use cases with regard to the Support-U application used to demonstrate the usefulness of the interdisciplinary patterns. The patterns themselves are outlined in Section 5. An evaluation of the usefulness of the patterns is provided in Section 6. Finally, the conclusion is given.

2 Support-U

In this section a brief introduction to the Support-U application is given. The application has been developed during the Venus project at the University of Kassel. Support-U is used to motivate and demonstrate the usefulness of the interdisciplinary development patterns outlined in Section 5 which can be used to increase the effectiveness of building socially acceptable software for ubiquitous systems. Support-U addresses the field of Ambient Assisted Living (AAL) and combines it with the field of ubiquitous computing to enable elderly persons to live autonomously. This issue becomes more and more important because the average life expectancy in the European Union has been increasing since 1990. Therefore, the number of elderly people who need care increases as well. Due to the fact that younger people have to be more and more flexible with respect to their working places most of them do live in a far distance to their elderly family members which makes a constant care difficult. Support-U addresses this challenge by providing e.g., a location-independent information window with direct view into the elderly person's appartment to family members. With the provided information window, family members can get an impression whether the elderly person is doing well or if she needs something, e.g. medical support and whether the conditions of her flat are satisfying. Vice versa, the elderly person can be sure that their family will notice emergencies that might occur to her. To provide this information window, the Support-U application consists of a ubiquitous environment that provides the information and an application that runs on a Tablet PC which visualizes the provided information. The ubiquitous environment, which is the flat of the elderly person, consists of various installed and unobtrusive sensors that gather current information used to infer contexts to describe the condition of the elderly person and the state of her flat. The need of people to get information about their family members who live far apart using mobile sensor technology is known as Social Link [DLK11]. Figure 1 outlines the start screen of the current version 3.0 of the Support-U application whereby different new interdisciplinary development patterns have been used during the development process to ensure the social acceptance of the application. Version 2.0 which has been developed without using the interdisciplinary patterns proposed in Section 5 has already been evaluated in [HSH+12]. The screen provides information to describe the condition of the elderly person by providing information of the person's blood pressure, pulse and her movement intensity. The accumulated person status is also outlined. Further the screen provides information that describes the status of the flat by showing activated electrical consumers e.g., the oven and the hotplates, opened windows or doors and the current temperature, humidity and light intensity in the flat.

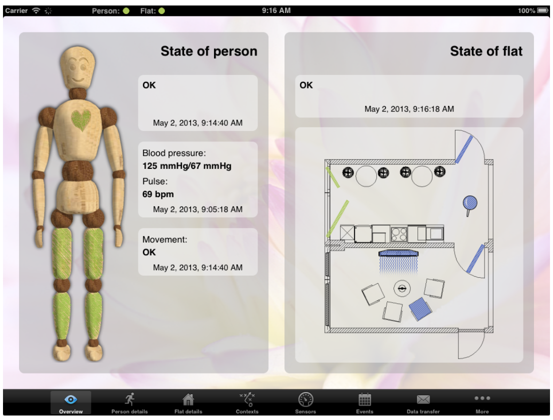

Figure 1: The main screen of the Support-U application. The left side outlines the condition of the person, the right side outlines the status of the flat.

3 Architecture

This section describes the underlying architecture that provides the user-based context data that is utilized by the Support-U application introduced in Section 2. In order to support elderly persons e.g., by their family members using the Support-U application, sensor data of the elderly person has to be collected, pre-processed, interpreted and provided to the application. This process is realized by the architecture outlined in Figure 2. As outlined in Section 4 the ubiquitous environment respectively the flat (kitchen and living room) the elderly person lives in, is equipped with heterogeneous types of sensors (1). FS20 sensors provide location data of the person or register whether an electrical consumer like the TV, the microwave, etc. is switched on. Phidget sensors are used to determine the current room conditions e.g., the temperature, the humidity and the light intensity. Further, they provide information whether the oven or the hotplates are turned on or turned off. Enocean sensors provide information whether the person is sitting on a certain chair. Finally, built-in sensors (gyroscope, accelerometer) of a smartphone the person carries are used to determine their current movement behavior (sitting, standing, walking) similar to the approaches presented in [LKD$^+$10a, LD10]. The sensor types use different modes to transmit their data. FS20 and enocean sensors use a radio-based transmission of their data, and the phidget sensors a cable-based transmission. All data of these different sensor types are first transmitted to a sensor client (2) that is used to push the data via HTTP using the REST-based interface [Fie00] of the so called context server (3). The sensor client enables a virtual connection between the FS20, phidget and enocean sensors with the context server. The smartphone is able to push its gathered sensor data directly to the context server via HTTP using the REST interface of the context server. The context server characterizes the main component of the proposed architecture. It basically consists of two different databases and sets up on an existing Java EE solution using the Spring framework. The graph-based database is realized using Neo4j[1]. By using the graph-based database relations between users living in the ubiquitous environment and their data that have been gathered using sensors installed in the same environment, can be established. The file-based database is realized using a python script that provides creates, reads, updates and deletes (CRUD) functionality, similar to an relational database application like SQL. The context server uses a file-based database because the number of writing accesses to store incoming sensor data, respectively interpreted or predicted high-level contexts are much higher than the number of reading accesses by applications e.g., Support-U that uses contexts for visualization or to adapt their behavior (5). After the heterogenous sensor data have been stored to the REST-based context server, context recognition approaches implemented as a service are used to interpret low-level contexts e.g. sensor data to high-level contexts (4). High-level context data represent a higher interpretation level of the context data, which is mostly related to a person or even characterizes a person in a ubiquitous environment. As algorithms to derive high-level context data from low-level or sensor data, data mining approaches such as the J48 classifier have been used. Furthermore, the framework provides context prediction approaches also implemented as services to forecast a user's next high-level contexts based on her previously sensed behavior patterns (4). One context prediction approach that is supported by the framework is the alignment predictor [SHD10]. An application example for using context recognition is the automatic recognition of the current movement behavior of the elderly person from the sensor data received by the smartphone. An application example to use context prediction is to forecast whether the person is cooking next based on her daily life routine. The recognized movement contexts are utilized by the Support-U application to inform the relative whether the person is sufficiently moving at this day and the predicted contexts are utilized to proactively inform the relative about the next action of the elderly person.

[1]http://www.neo4j.org/

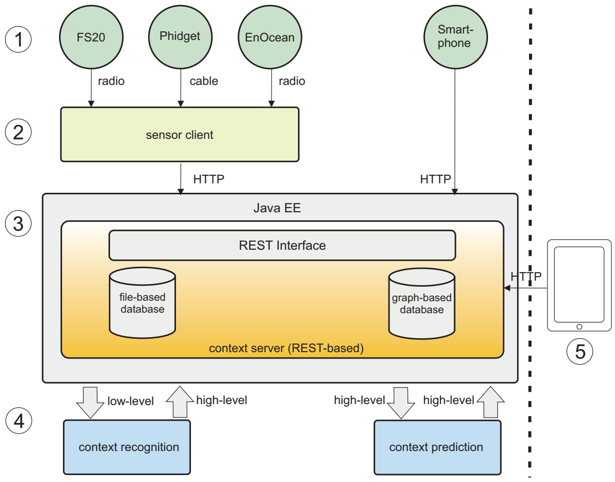

Figure 2: Architecture to process and distribute context aware sensor data.

4 Use Cases

In this section, two selected use cases named ”proactive danger recognition” and ”context detection” are outlined to motivate the usefulness of the interdisciplinary patterns that have been used during the development process of Support-U to increase its social acceptance by the users. The use cases are divided into the following four different segments: name of the use case, its motivation, a description of the use case and the resulting consequences for the user. The consequences describe the affects for the user that have to be addressed by the patterns outlined in Section 5 to ensure social acceptability.

4.1 Proactive danger recognition

Providing proactiveness is one of the most common features of context aware applications in ubiquitous environments and systems. By the usage of proactiveness context aware applications can be enabled to be adapted to the users in advance by anticipating their needs. Examples are given in [MRT04, NMF05].

Motivation The motivation of this use case is to enable persons, who are in charge to support an elderly person using Support-U, to be warned proactively with respect to possible dangerous situations. Possible dangerous situations might occur with regard to the constitution of the elderly person or with regard to the status of the flat. A possible dangerous situation can be the cooking process if the elderly person often forgot to switch off the hotplates, in the past. Therefore, a proactive warning can be used to provide a certain time advantage for the person in charge, which enables the person to be more sensitized with respect to possible upcoming complications that might occur during or after the cooking process.

Description To enable context aware applications to react or adapt proactively according to a person's following context, context prediction techniques like e.g. the alignment approach [SHD10] is used (cf. Figure 2). These context prediction techniques utilize a user's past contexts that represent e.g. her daily life routine and her current context information e.g. the current action of a person to predict the person's possible next context.

Consequences To enable context prediction approaches like Alignment or the Collaborative-based Context predictor to work reliably they need a high number of already gathered contexts that describe the actions of a person in the past. For this reason context data sensed by sensors installed in a ubiquitous environment have to be stored permanently. Further the data is mostly stored externally by the service provider which offers the prediction service to be easily accessed. For this reason, a person loses control of her own personal data if she uses a prediction service to get proactively informed about possible dangerous situations.

4.2 Context detection

The utilization of mostly personal context information by context aware application is a crucial condition for its effective usage. Contexts are derived from gathered sensor data and are used to adapt or even proactively adapt a context aware application to the user's needs. A good overview of existing approaches in context awareness is provided in [PB03].

Motivation The motivation of this use case is to recognize context information (cf. Figure 2) that characterizes the conditions of the elderly person and the status of her flat. The recognized context information is further visualized by the Support-U application that is used by the person who supports the elderly person to live autonomously. Visualized contexts can be e.g., switched electronically devices or the movement behavior of the elderly person.

Description To provide useful context data that characterizes the elderly person or the status of the flat a high number of sensors are needed that are installed all over the flat of the elderly person. Support-U visualizes contexts derived from approximately 30 different sensors. The idea of ubiquitous computing is to be unobtrusively and do not impair or distract the user in her normal behavior. This implies that the installed sensors are not visible to the elderly person who behaves in the ubiquitous environment.

Consequences Since the installed sensors are unobtrusive to the user it is not obvious for the elderly person or the person who supports the elderly person using Support-U which sensors are installed in the environment and what kind of personal data the sensors are currently sensing. This implies a high level of trust by the users.

5 Interdisciplinary Patterns

Patterns are used in a broad spectrum during the development process of software. They are used to describe, to optimize and to present a general reusable solution for commonly occurring problems in a generic way. Basically there are four different types of patterns that are used to structure the development process of software. Well-known are software design patterns described by the Gang of Four [GHJV95]. Software design patterns are used to speed up the development process of software by providing standardized development paradigms that have already been successfully applied and tested in many software development projects. A second pattern type is represented by idioms. Idioms describe the implementation of specific tasks e.g., the implementation of an algorithm using a concrete programming language. Therefore idioms provide a more concrete problem solution than software design patterns [Cop92]. Another type of patterns are architectural ones first described in [BMR+96]. Contrary to software design patterns or idioms that describe concrete problem solutions to local and small problems, architectural patterns are used to provide a description of the organization and interaction of parts of an application. Examples for an architectural pattern are Peer-to-Peer or Client-Server communication models that support the usage of distributed resources and services in distributed systems. Patterns that provide conceptual models in software engineering are so called software analysis patterns. Software analysis patterns give an overview of complex connections between objects in a given problem space. Examples are outlined in [Fow96].
The patterns that have been identified during the interdisciplinary development process of the Support-U application will be outlined next. The patterns are described using a standard template that specifies the intent of the pattern, the motivation of the pattern, possible forces and the context of the pattern, the solution of the pattern and a short discussion of the consequences that might occur while using the pattern. The structure of this template has already been successfully outlined and used in [GHJV95, BMR+96]. All patterns that will be described are not disciplinary but affect multiple disciplines in the development process. These disciplines are related to information technology, to usability, to trust or to legal regulations.

5.1 Informational Self-determination Pattern

Intent The goal of the pattern is to enable the user to explicitly agree or disagree to certain functionalities provided by a context-aware application. This possibility to enable or disable certain functionality has to be provided to the user by the application at it's first start.

Motivation Ubiquitous computing systems often gather personal sensor data respectively derive personal context data of the users that utilise these systems. These systems consider users that e.g., live in Ambient Assisted Living environments. These users are surrounded by various sensors. Examples are video cameras, motion sensors or sensors used to recognise whether an electrical consumer has been switched on. All these sensors can be used to receive information describing the current situation of a person. Another example can be given by ubiquitous computing systems that apply a user's movement behaviour, which is sensed by a acceleration sensor built-in in modern smartphones. These movement contexts are used to recommend places of interests. With regard to these examples it becomes obvious that sensors often unobtrusively collect highly critical and personal context data of the concerning users. Hence, the proposed pattern enables the user to decide which functionality she is willing to use and which functionality the user renounces because she does not want to provide the context data needed by the required functionality.

Forces and Context

- Informational self-determination: The pattern considers a user's right of informational self-determination. The reason is, that a user is able to explicitly agree or disagree to a certain functionality depending on the context data needed by the functionality. Therefore, the user has direct control over the context data collection process. This satisfies the principles (necessity, transparency, giving consent and responsibility [Kun07] at pages 63 - 108 and [HS09] at page 87) of a user's right of informational self-determination.

- Trust: The pattern encourages a user's trust by offering the possibility to prevent the collection and inference of certain personal context data. Hence, a user can be sure that personal data that is critical to her is not gathered, stored or even further processed by third parties.

- Transparency: The pattern provides transparency to the user by providing an overview, which personal context data is needed by which functionality to work properly. For this reason a user can be aware of the context data that is gathered by the sensors that surround her.

Solution A solution is given if the user can explicitly agree or disagree to certain functionalities. This can be provided by nearly every ubiquitous computing system by naming all functionalities and all sensors respectively context data utilized by a functionality. A possibility to display these functionalities and the used context data is to use the privacy consent form, which is included in every application. Figure 3 displays the privacy consent for of the Support-U application. In the displayed privacy consent form each functionality, which utilises personal context information is listed. Further, the user is able to agree or to disagree to the functionalities. Functionalities offered in Support-U that use personal context information are video streaming and the prediction of a user's next contexts based on her previously sensed contexts.

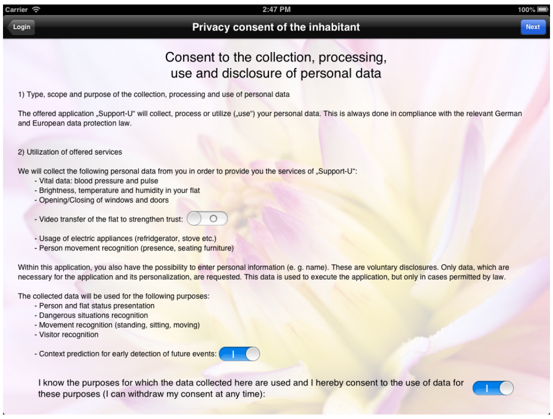

Figure 3: Example of the usage of the self-determination pattern. The pattern was applied in the data consent form of the Support-U application.

Consequences By enabling the user to explicitly agree or to disagree to certain functionalities a context aware application e.g. Support-U might not be able to provide all of its possible functionalities to the user anymore.

5.2 TrustParency Pattern

Intent The intent of the TrustParency pattern is to visualize hardware sensors or even inferred context information that surrounds a user, and that is used by a ubiquitous computing system. Further, the idea of the TrustParency pattern is to reduce the unobtrusiveness of current sensor technology if needed.

Motivation To the best of our knowledge gathered sensor data and inferred user contexts are mostly provided to the user without visualizing the sensors used by a ubiquitous computing system. These, mostly highly personal context data is used to provide adaptability and proactiveness of these ubiquitous computing systems. Various sensors pervade our daily life and affect us in different situations and areas. In the field of health care, possibilities were elaborated to give patients the opportunity to be monitored even if they are outside of a hospital using ubiquitous sensors built-in smartphones [LKD+10b]. So called smart homes and smart rooms adapt their services to the lifestyle habits of occupants and the working routines of clerks by observing and learning their behaviour patterns [DS08]. The automotive application domain represents another area, which is strongly influenced by ubiquitous sensors. Sensors such as infrared, radar, laser or GPS sensors are e.g., used to prevent possible collisions between a car and a pedestrian [DF10]. With the aid of smart badges conference attendees can be grouped by their interests. They can be automatically informed about similar activities of other members [PGL+10]. Further, RFID sensors can be used to detect whether conference attendees are talking to each other, how long their conversation took, and which talks participants have visited to provide them with additional information. This information can e.g., be other interesting talks at the conference or other attendees with similar interests based on a user's profile [ABD+11].

Forces and Context

- Trust: The pattern encourages a user's trust in using ubiquitous computing system by visualizing the context sources, respectively the sensors that surrounds the user [SHHL12, SHH+12].
- Transparency: The pattern provides transparency from a legal perspective, not from a technical perspective. Hence, the user gets the possibility for a better understanding of her environment by getting transparency. This means, a user is enabled to see what sensor technology surrounds her in her daily environment.

Solution In general a solution can be the usage of an augmented reality view in a context-aware application. This augmented view can be e.g., provided using the camera of a smartphone. Using the augmented reality view, sensors that surrounds the user can be visualized. With regard to the Support-U application, which offers a view directly in the flat of an elderly person an overlay functionality is proposed to visualize the sensors installed in the ubiquitous environment of the elderly person. These overlays enable the user to easily recognize the type of sensor that is installed. Further, the user is able to access the current data of the sensor and the data history of the sensor by simply clicking the overlay that represents the sensor the user is interested in. An example of a possible implementation is given in Figure 4.

Figure 4: All unobtrusive sensors that are installed in the ubiquitous environment are visualized using overlays.

Consequences By enabling the user to see what technical sensors surrounds her, her confidence can be encouraged and transparency is provided. Further, the user is able to access additional information that has not been visible and accessible to her before.

5.3 Discussion of the outlined patterns

The interdisciplinary patterns that have been outlined in this section present a selection out of a group of five patterns that have been discovered and applied to the development process of the current version of Support-U. Just like the presented patterns the other three patterns can be used to increase the social acceptability of context aware applications with respect to usability, trust and legal regulations. In the following the other three patterns are shortly outlined for the sake of completeness, without following the structure used to describe the Self-determination and the TrustParency pattern.

- *One-Click pattern*: Enables the user to navigate from each screen to any other screen by only one click. This pattern prevents the user to get lost in the depths of context aware applications.
- *TaC-Short pattern*: Provides a short version of the terms and condition that summarizes the most important points to a user exactly using only one page of the screen. The long version is also provided.
- *Highlight pattern*: Enables the user to easily classify the context information presented in a context aware application. Thereby, a color scheme consisting of blue, green and red color is used. Blue highlights neutral context information, green highlights significant contexts whose current status are positive to the user, red highlights significant contexts whose current status are negative to the user.

The two patterns that have been outlined in detail have been utilized to overcome the identified consequences of the two use cases described in Section 4. The Self-determination pattern addresses the consequences of the use case ”proactive danger recognition” and the TrustParency pattern addresses the consequences of the use case ”context detection”. By utilizing the Self-determination pattern it can be ensured that the user can decide whether she wants to provide her personal contexts to context prediction processes to enable the proactively warning functionality, or not. If the user declines the functionality the user can explicitly prevent the system form storing her personal data externally. Therefore, she does not lose control of her personal data. The possibility for the user to comply or to not comply with certain functionality is provided directly in the data consent form. For this reason, it can be ensured that the user can take a decision before the user utilizes the context aware application e.g., Support-U for the first time. If the user changes her mind the user should be able to activate or deactivate the functionality later. With the help of the Self-determination pattern the user receives the opportunity to use Support-U and any other context aware applications even if the user did not a agree to all functionalities of the application. The application of the TrustParency pattern enables the user to receive information about the sensors installed in the ubiquitous environment that surrounds her. For this reason, the presented pattern primarily supports the transparency. Thereby, it is not about the transparency from a technical point of view but it is about the transparency from a legal point of view. Hence, a user is enabled to understand the system that utilizes her personal contexts. By enabling transparency the TrustParency pattern encourages the trust of the user in using context aware application. Transparency is provided by Support-U and can be provided by other context aware applications by utilizing so called overlays that visualize the installed sensors by providing information about the sensor type and its position in the ubiquitous system. Both, the Self-determination and the TrustParency pattern that have been presented and shortly discussed show possibilities to boost the social acceptance of the persons that use context aware applications. The current version 3.0 of Support-U has already been developed according to these patterns. An evaluation how the utilization of these interdisciplinary patterns actually affects the social acceptance of the user is given in the next Section. Thereby, version 3.0 of Support-U is opposed to version 2.0 that has been developed without the presented interdisciplinary patterns.

6 Evaluation

To evaluate the usefulness of our patterns, we conducted a structured evaluation comparing Support-U version 2.0 and 3.0. We invited eight student participants, which used both versions of Support-U. The evaluation was guided by an author of the paper to ensure that the participants came in touch with the whole functionality of both versions. After using both versions, the participants were asked several closed and open questions about Support-U. The average evaluation took about 45 minutes including 30 minutes of using both versions and 15 minutes of answering questions. The goal of the evaluation was on the one hand to gather suggestions regarding how Support-U could be further improved, and on the other hand to get feedback on the usefulness of the functionalities connected to each pattern that has been used for developing version 3.0 of Support-U. In the remainder of this section, we will focus on presenting the feedback regarding the patterns, since the patterns are the focus of this paper. First, we asked the participants which version of Support-U they perceive as being better. Here, six of the eight participants preferred version 3.0, one participant preferred version 2.0 and another participant was indifferent. So on an overall level, we can conclude that Support-U has improved from version 2.0 to version 3.0. However, this result does not allow us to draw any conclusion on the usefulness of the patterns presented in this paper. Consequently, we further asked the participants questions about the usefulness of specific functionalities. All functionalities were related to the presented patterns, but this was unknown to the participants. We first asked the participants to provide open feedback on each functionality, and, afterwards, they were asked to rate the importance of the existence of each functionality on a 5-point Likert response format (1 = not important at all, 5 = very important). We used the results of this rating process to develop a ranking of patterns based on how important the resulting functionality was rated by the participants of our evaluation (cf. Table 1). The results presented in Table 1 show that the functionality related to the Self-determination pattern was ranked most important, followed by the

Table 1: Ranking results of the interdisciplinary patterns rated by eight participants.

	Self-determination pattern	Trust-Parency pattern	One-Click pattern	TaC-Short pattern	Highlight pattern
User 1	5	4	4	3	2
User 2	5	2	5	5	1
User 3	5	5	5	5	4
User 4	4	4	3	3	4
User 5	5	5	4	5	3
User 6	5	5	2	4	1
User 7	4	4	3	3	2
User 8	3	5	4	2	1
Sum	36 / 40	34	30	30	18
Rank	1	2	3	3	5

functionality related to the TrustParency pattern. The functionalities related to the One-click and the TaC-Short patterns were ranked third. Regarding the functionality related to the Highlight pattern, we observed a low value (well below the value of 24, which would resemble medium importance). As a result, we can conclude that four of our five provided patterns will lead to the implementation of functionalities which are considered important or even very important by potential users.

7 Conclusion

In this paper we outlined interdisciplinary patterns to provide reusable solutions for common problems that affect the development of context aware application used in ubiquitous environments. Problems of context aware application are often raised by the implicit usage of personal context data belonging to the user to adapt or to proactively adapt the application to the user's needs. For this reason, the proposed interdisciplinary patterns address social related implications with regard to transparency, trust and self-determination that can be raised by the implicit usage of personal context data rather than technical issues. Two patterns the ”TrustParency pattern” and the ”Self-determination pattern” were presented in more detail. For the demonstration of the interdisciplinary patterns, Support-U, a context aware application that provides elderly people to live autonomously, is used. The proposed patterns were exemplarily applied to two use cases of the Support-U application named ”proactive danger recognition” and ”context detection”. It could be shown that the ”TrustParency pattern” and the ”Self-determination pattern” can be used to overcome the identified consequences to the user of the two use cases. Further, the proposed patterns were evaluated by eight students. The participants provided feedback to the usefulness of the functionalities connected to the patterns. The results showed that the functionality related to the Self-determination and the TrustParency pattern was ranked most important.

8 Acknowledgement

The authors are involved in the VENUS research project. VENUS is a research cluster at the interdisciplinary Research Center for Information System Design (ITeG) at Kassel University. We thank Hesse's Ministry of Higher Education, Research, and the Arts for funding the project as part of the research funding program "LOEWE - Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz".

References

[ABD+11] Martin Atzmüller, Dominik Benz, Stephan Doerfel, Andreas Hotho, Robert Jäschke, Björn-Elmar Macek, Christoph Scholz Mitzlaff, Folke, and Gerd Stumme. Enhancing Social Interactions at Conferences. *it - Information Technology, (53)3:101-107, 2011*, 2011.

[BMR+96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[Cop92] James O. Coplien. *Advanced C++ programming styles and idioms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[DF10] K. David and A. Flach. An innovative Car-2-X system concept for pedestrian safety. *IEEE VTC Journal*, pages 70–76, mar 2010.

[DJ12] K. David and N. Jefferies. Wireless Visions: A Look to the Future by the Fellows of the WWRF. *Vehicular Technology Magazine, IEEE*, 7(4):26–36, 2012.

[DLK11] Klaus David, Sian Lau, and Bernd Klein. Social Link APP. *IEEE Vehicular Technology Magazine*, 6(3):80–87, September 2011.

[DS08] Maria Danninger and Rainer Stiefelhagen. A context-aware virtual secretary in a smart office environment. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 529–538, New York, NY, USA, 2008. ACM.

[Fie00] Roy Thomas Fielding. *REST: Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.

[Fow96] Martin Fowler. *Analysis Patterns: Reusable Object Models*. Addison-Wesley Professional, 1 edition, October 1996.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[HS09] G. Hornung and C. Schnabel. Data protection in germany: The population census decision and the right to informational self-determination. 25(1):84–88, 2009.

[HSH+12] Sebastian Hoberg, Ludger Schmidt, Axel Hoffmann, Matthias Söllner, Jan Marco Leimeister, Christian Voigtmann, Klaus David, Julia Zirfas, and Alexander Roßnagel. Socially acceptable design of a ubiquitous system for monitoring elderly family members. In *Sozio-technisches Systemdesign im Zeitalter des Ubiquitous Computing (SUBICO 2012)*, pages 349–363, 2012.

[IR09] M. Iqbal and M. Rizwan. Application of 80/20 rule in software engineering Waterfall Model. In *Information and Communication Technologies, 2009. ICICT '09. International Conference on*, pages 223–228, 2009.

[Kun07] Christopher Kuner. *European Data Protection Law, Corporate Compliance and Regulation*. Oxforf University Press, 2007.

[LD10] Sian Lun Lau and Klaus David. Movement recognition using the accelerometer in smartphones. In *Future Network and Mobile Summit, 2010*, pages 1–9, Florence, Italy, 2010. IEEE.

[LKD+10a] Sian Lun Lau, Immanuel. König, Klaus David, Baback Parandian, C. Carius-Dussel, and Martin Schultz. Supporting patient monitoring using activity recognition with a smartphone. In *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, pages 810–814, 2010.

[LKD+10b] Sian Lun Lau, Immanuel König, Klaus David, Baback Parandian, Christine Carius-Düssel, and Martin Schultz. Supporting Patient Monitoring using Activity Recognition with a Smartphone. In *The Seventh International Symposium on Wireless Communication Systems (ISWCS'10)*, York, UK, 2010.

[Man] Steve Mann. My "Augmediated" Life - IEEE Spectrum. http://spectrum.ieee.org/geek-life/profiles/steve-mann-my-augmediated-life.

[MRT04] Kari Laasonen Mika, Mika Raento, and Hannu Toivonen. Adaptive On-Device Location Recognition. In *In Proceedings of the Second International Conference on Pervasive Computing*, pages 287–304. Springer Verlag, 2004.

[NMF05] Petteri Nurmi, Miquel Martin, and John A. Flanagan. Enabling proactiveness through context prediction. In *Proceedings of the Workshop on Context Awareness for Proactive Systems, Helsinki*, volume 53, 2005.

[PB03] Paul Prekop and Mark Burnett. Activities, context and ubiquitous computing. *Comput. Commun.*, 26(11):1168–1176, July 2003.

[PGL+10] Joseph A. Paradiso, Jonathan Gips, Mathew Laibowitz, Sajid Sadi, David Merrill, Ryan Aylward, Pattie Maes, and Alex Pentland. Identifying and facilitating social interaction with a wearable wireless sensor network. *Personal and Ubiquitous Computing*, 14(2):137–152, February 2010.

[SHD10] S. Sigg, S. Haseloff, and K. David. An Alignment Approach for Context Prediction Tasks in UbiComp Environments. *Pervasive Computing, IEEE*, 9(4):90–97, December 2010.

[SHH+12] Matthias Söllner, Axel Hoffmann, Holger Hoffmann, Arno Wacker, and Jan Marco Leimeister. Understanding the Formation of Trust in IT Artifacts. In *Proceedings of the International Conference on Information Systems (ICIS)*, Orlando Florida, USA, 2012.

[SHHL12] Matthias Söllner, Axel Hoffmann, Holger Hoffmann, and Jan Marco Leimeister. Vertrauensunterstützung für sozio-technische ubiquitäre Systeme. *Zeitschrift für Betriebswirtschaft*, 2012(Supplement 4):109–140, 2012.

[SL08] Liu Shuping and Pang Ling. The Research of V Model in Testing Embedded Software. In *Computer Science and Information Technology, 2008. ICCSIT '08. International Conference on*, pages 463–466, 2008.

[Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, September 1991.